Refine Search

Search Results -

Term	Documents
ADDRESS	265938
ADDRESSES	155142
TABLE$	0
TABLE	861464
TABLEA	23
TABLEACTION	1
TABLEACTIVATE	2
TABLEADD	1
TABLEADDRESS	3
TABLEADDRESS+63	1
TABLEADDRPTR	1
((L8 OR L7) AND ((ADDRESS ADJ1 TABLE$) OR (ADDRESS ADJ1 MAP$))).USPT.	7

There are more results than shown above. Click here to view the entire set.

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L10

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Tuesday, February 01, 2005 Printable Copy Create Case

Set Name **Query** **Hit Count** **Set Name**
side by side result set

DB=USPT; PLUR=YES; OP=ADJ

h e b b cg b e e ch

L10: Entry 4 of 7 File: USPT Jul 15, 2003

DOCUMENT-IDENTIFIER: US 6594776 B1
TITLE: Mechanism to clear MAC address from Ethernet switch address table to enable network link fail-over across two network segments

Abstract Text (1):
There is provided a communication network and method for enhancing server availability to client PCS which includes two Ethernet switches. Each one of the two Ethernet switches is connected to a corresponding one of the primary and secondary network interface cards in the file server PC. The two Ethernet switches are interconnected together through an uplink port. As a result, redundancy has been effectively and efficiently provided against the failure of either one of the two switches in order to enable link fail-over across two network segments.

Brief Summary Text (11):
It is an object of the present invention to provide a mechanism and method for clearing MAC address from Ethernet switch address table so as to enable network link fail-over across two network segments.

Brief Summary Text (13):
In a preferred embodiment of the present invention, there is provided a communication network for enhancing server availability to client PCS which includes a file server PC having a primary network interface card and a secondary network interface card configured as a fail-over pair. The primary and secondary network interface cards are programmed with a single MAC address. A first switch device is used to store initially the MAC address of the server PC in an address table. The first switch device is coupled to the primary network interface card via a primary link.

Detailed Description Text (4):
In the normal operating condition, the Ethernet switch A (124) has initially stored in its address table the MAC address of the primary NIC. Therefore, the client PC 112a, for example, is able to be connected to the file server PC 114. In the event that the primary link 118 connected to the primary NIC should malfunction or fail, such as the cable or link being disconnected and/or the switch 124 failing, the secondary NIC will then take over. Further, the switch A (124) will delete the MAC address of the server PC 114 from its address table. As a result, the secondary NIC will send an LLC broadcast packet to the switch B (132).

Detailed Description Text (5):
Thus, the default or secondary Ethernet switch B (132) will add the server=s MAC address to its address table. Now, when the client PC 112a wants to send a packet to the file server PC 114, the primary Ethernet switch A (124) will not be able to find the MAC address in its table. As a consequence, the primary switch A will flood all of the ports as well as the uplink port 134. In this manner, unlike the prior art having a single Ethernet switch, the packet sent by the client PC 112a can then be received by the second or backup Ethernet switch B (132) in order to forward the same to the file server PC 114. Therefore, the client PC 112a is again able to be connected to the server PC 114.

Detailed Description Text (6):
Now if the primary link or cable 118 is re-connected or reinstated and/or the primary switch 124 is restored, the server PC software will switch control from the secondary NIC to the primary NIC. However, due to the fact that the secondary NIC remains connected to the backup Ethernet switch B (132) through the secondary link or cable 128, the backup switch B would still continue to have the MAC address of the server PC 114 stored in its address table. As a result, if the client PC 112a were connected to the backup switch B (132) it would not be able to be connected to the server PC 114 due to the fail-back process of the primary NIC. In view of this, when the server PC 114 completes the fail-back, the secondary NIC is also used to break the secondary link or cable 128 for a short period of time. In this fashion, the backup switch B will cause the MAC address of the server PC 114 to be deleted from its address table. As a consequence, the client server 112a will again be able to be connected to the server PC 114 via the primary switch A and the primary link 118.

Detailed Description Text (7):
In FIG. 3, there is depicted a flow chart 300 which illustrates the fail-over process where the secondary switch B becomes active when there is a failure in the primary link/switch A of FIG. 2. The process begins in the Start block 302 where there is an initialization of the network interface cards (NIC) and their associated drivers in the server PC 114. Initially, the primary NIC coupled via the driver and a physical layer to the primary link 118 is active, and the secondary NIC coupled to the secondary link 128 does not transmit or receive any frames. Since the primary NIC and the secondary NIC share a common MAC or physical address, the presence of the secondary NIC is hidden from the client PCS 112a-112e. Therefore, when the switch A (124) starts to receive frames from the primary NIC via the primary link. 118, it will add the MAC address to its address table.

Detailed Description Text (8):
In block 304, the driver monitors periodically the status of the primary link 118 in order to determine whether there is a failure in the primary NIC, primary link, or switch A. If the answer is "NO", then the process goes to the block 306 where the primary NIC is continued to be used for sending and/or receiving of the frames and is looped back to the block 304. If the answer is "YES" from the block 304, the process will proceed to block 308 in which the driver initiates a fail-over process by transferring control to the secondary NIC. Upon finding that there is a failure in the primary link 118 to the primary NIC, the primary switch A in the block 310 will remove the NIC's address from its address table.

Detailed Description Text (9):
Once the fail-over process has been completed and the secondary NIC is ready to take over the network traffic from the primary NIC, the driver in block 312 will send a broadcast LLC frame using the secondary NIC to the secondary link 128 and the switch B. In the block 314, when the switch B receives this LLC frame, it will add the MAC address to its address table. In block 316, the driver will continue to monitor periodically the status of the primary link 118 in order to determine whether the primary NIC is back on-line. If the answer is "NO", then the process goes to the block 318 where the secondary NIC is continued to be used for sending and/or receiving of the frames and is looped back to the block 316. If the answer is "YES" from the block 316, the process will proceed to block 322 in which the driver initiates a fail-back process by transferring control to the primary NIC.

Detailed Description Text (10):
Upon finding that the failure has been repaired, such as re-connecting of a disconnected cable or replacing the failed NIC with a new one (i.e., "Hot Swap" procedure), in the block 322 the link pulses being transmitted from the secondary NIC are then turned off for a short period of time which is accomplished by resetting a device in the physical layer. Since the device in the physical layer requires a certain amount of time before re-initialization, the link pulses will be

turned off during this time interval. This causes the secondary switch B to assume that the secondary link 128 has failed. As a result, the secondary switch B in block 324 will remove the NIC's address from its address table.

Detailed Description Text (11):
Thereafter, the driver in block 326 will again send a broadcast LLC frame using the primary NIC to the primary link 118 and the switch A. In the block 328, when the switch A receives this LLC frame, it will add the MAC address to its address table again. Since the LLC (Logical Link Control) frame is broadcasted, the switch A (124) will forward the LLC frame to the switch B (130) via the uplink port 134. This causes the switch B to associate the server PC's MAC address with the uplink port 134 in its address table. As a result, if a client PC should be connected to the switch B, then such client PC would be caused to be connected to the server PC 114 via the uplink port 134, the primary switch A, and the primary link 118. The fail-back process is completed in the End block 330. However, the overall process is looped back to the block 304 as indicated by the line 332 in order to repeat the same.

Detailed Description Text (12):
From the foregoing detailed description, it can thus be seen that the present invention provides a mechanism and method for clearing the MAC address from the switch address table. This is achieved by the provision of two Ethernet switches each being connected to a corresponding one of the primary and secondary network interface cards in the file server PC. The two switches are interconnected together through an uplink port. As a result, redundancy is effectively provided against failure of either one of the two Ethernet switches so as to enable linked fail-over across the two network segments.

CLAIMS:

1. A communication network for enhancing server availability to client PCS, said communication network comprising: a file server PC having a primary network interface card and a secondary network interface card configured as a fail-over pair, said primary and secondary network interface cards being programmed with a single MAC address; a first Ethernet switch for initially storing the MAC address of said server PC in an address table; said first Ethernet switch being coupled to the primary network interface card via a primary link; a plurality of client PCS being disposed in communication with said first Ethernet switch; one of said plurality of client PCS being in communication with said server PC through said first Ethernet switch in normal operation; a second Ethernet switch for functionally replacing said first Ethernet switch when there is a failure in the first Ethernet switch and/or the primary link; said second Ethernet switch being coupled to the secondary network interface card via a secondary link; an uplink port for interconnecting said first and second Ethernet switch so as to enable link fail-over from one of said plurality of client PCS through said second Ethernet switch to said file server PC upon occurrence of the failure; the MAC address stored in said first Ethernet switch being deleted upon occurrence of the failure and said secondary network interface card sending an LLC broadcast packet to said second Ethernet switch to cause the MAC address to be added in an address table in said second Ethernet switch so as to allow said one of said plurality of client PCS to be connected to the server PC via said uplink port, said second Ethernet switch, said secondary link, and said secondary network interface card; and said primary network interface card taking back control upon restoration of said first Ethernet switch and/or primary link that failed and said second Ethernet switch breaking said secondary link for a short period of time to cause the MAC address to be deleted from the address table of said second Ethernet switch, thereby providing a fail-back link from the said one of said plurality of client PCS to the server PC via said first Ethernet switch, said primary link, and said primary network interface card.

Previous Doc Next Doc Go to Doc#

Refine Search

Search Results -

Term	Documents
REDUNDANCY	45928
REDUNDANCIES	2512
REDUNDANCYS	0
(3 AND REDUNDANCY).USPT.	2
(L3 AND REDUNDANCY).USPT.	2

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L4

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Tuesday, February 01, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L4	L3 and redundancy	2	L4
L3	L1 and (dynamic$ with map$ with virtual with address$)	5	L3
L2	L1 and (dynamic$ with map$ with virtual with address$).ab.	0	L2
L1	(network$ and communicat$).ab.	16476	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Term	Documents
VIRTUAL	68568
VIRTUALS	8
DYNAMIC$	0
DYNAMIC	229878
DYNAMICA	13
"DYNAMICACCESE.RTM"	1
DYNAMICACCESS	2
"DYNAMICACCESSE.RTM"	1
"DYNAMICACCESS.RTM"	4
DYNAMICAIRE	1
DYNAMICAL	2253
(L1 AND (DYNAMIC$ WITH MAP$ WITH VIRTUAL WITH ADDRESS$)).USPT.	5

There are more results than shown above. Click here to view the entire set.

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L3

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Tuesday, February 01, 2005 Printable Copy Create Case

Set Name **Query** **Hit Count** **Set Name**
side by side **result set**

DB=USPT; PLUR=YES; OP=ADJ

h e b b cg b e e ch

L4: Entry 1 of 2 File: USPT Feb 3, 2004

DOCUMENT-IDENTIFIER: US 6687735 B1
TITLE: Method and apparatus for balancing distributed applications

Abstract Text (1):
An improved method and apparatus for balancing distributed applications within a client/server network, such as a cable television network, is disclosed. In one aspect of the invention, a method of balancing the load of distributed application client portions (DACPs) across various server portions (DASPs) and server machines is disclosed. Statistics are maintained by one or more software processes with respect to the available resources of the servers and their loading; new process threads and/or distributed application server portions are allocated across the servers to maintain optimal system performance as client device loading increases or changes. In another aspect of the invention, a novel object-oriented distributed application software architecture employing both vertical and horizontal partitions and "mutable" (i.e., transportable) objects is disclosed. The mutable objects may reside on either the server or client portions of the distributed application while maintaining at least one network partition. A runtime environment adapted for the operation of the foregoing object-oriented distributed application, including an efficient message protocol useful for interprocess communication, is also disclosed. Methods for downloading the DACP from the servers, and scaling the DACP at download based on client device configuration, are further disclosed.

Detailed Description Text (72):
The message protocol (MP) of the invention further assigns virtual addresses (VAs) to DASPs and DACPs, so that distributed application portions can move dynamically within the distributed application balancing system network. Servers associated with the distributed application balancing system network have records, for example, in their respective distributed application balancing system databases 706 that contain this dynamic mapping of virtual addresses. Clients on the network are only given those virtual addresses necessary to their communications needs. In one embodiment, however, clients can discover the virtual address of other DASPs and DACPs by sending a query message to the server farm 708. Discovery of such virtual addresses may be performed for, inter alia, identifying a well known server that provides a specific service, or to find applications of the same type running on other client devices.

Detailed Description Paragraph Table (1):
TABLE 1 Field name Size Description Flags 1 byte Contains multiple fields that imply the format of the rest of the message: Number of Bit 7 of the If on, indicates that the number of buffers greater Flags field buffers in the message is greater than than 1 1. If off, the number of buffers is equal to 1. Acknowledge Bit 6 of the If on, indicates that the message is an Flags field acknowledgement to the sent message indicated by the session ID and transaction ID fields. Not Bit 5 of the If on, indicates that the message Acknowledge Flags field recipient detected a problem with the sent message indicated by the session ID and transaction ID fields. The sender should re-send the message. Reply Bit 6 and Bit 5 If both bits are on, indicates that this of the Flags field message is a reply to the sent message indicated by the session ID and transaction ID fields. Version Bits 0 through 4 Indicates the version of the message number of the Flags field format.

Values can go from 0 to 35 and will wrap back to 0 at 36. Command 2 bytes Indicates the command type that this Type message should invoke on the recipient machine. The commands 0 through 255 are reserved for system commands, the other values can be used for application specific commands. Session Id 1 byte Indicates the session number given to the application by the server via the open session reply message. Transaction Id 1 byte A number assigned to the current transaction. Each transaction consists of multiple messages with a send a reply and acknowledgements being typical. Sent time 3 bytes Time that the message was sent. Encryption Bit 7 of the most If on, the message is encrypted. The significant byte encryption algorithms, and data in the Sent time format are specified between each field. DASP and DACP using the start, ("open session command"). Cyclic Bit 6 of the most If on, indicates that the last byte in Redundancy significant byte the message is an 8 bit cyclic Check (CRC) in the Sent time redundancy code (CRC). field. Minutes Bits 0 through 5 System time minutes of the time the in the most message was sent. significant byte of the Sent time field. Seconds Bits 2 through System time seconds of the time the eight in the message was sent. second most significant byte in the Sent time field. Milli-seconds Bits 0 through 8 System time milli-seconds of the time in the least the message was sent. significant byte and bits 0 and 1 in the second most significant byte in the Sent time field.

Previous Doc Next Doc Go to Doc#

US006687735B1

(12) **United States Patent**
Logston et al.

(10) **Patent No.: US 6,687,735 B1**
(45) **Date of Patent: Feb. 3, 2004**

(54) **METHOD AND APPARATUS FOR BALANCING DISTRIBUTED APPLICATIONS**

(75) Inventors: **Gary Logston**, Poway, CA (US); **Patrick Ladd**, San Marcos, CA (US)

(73) Assignee: **Tranceive Technologies, Inc.**, Carlsbad, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 518 days.

(21) Appl. No.: **09/583,064**

(22) Filed: **May 30, 2000**

(51) **Int. Cl.**[7] **G06F 15/16**

(52) **U.S. Cl.** **709/203**; 709/217; 709/219; 709/220; 709/316; 709/328; 709/329; 370/486; 370/487

(58) **Field of Search** 709/203, 217, 709/219, 220, 316, 328, 329; 370/486, 487

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,969,092 A	11/1990	Shorter	
4,991,089 A	2/1991	Shorter	
5,457,797 A	10/1995	Butterworth et al.	
5,555,244 A	9/1996	Gupta et al.	
RE35,448 E	2/1997	Shorter	
5,606,493 A	2/1997	Duscher et al.	
5,664,093 A	9/1997	Barnett et al.	
5,673,265 A	9/1997	Gupta et al.	
5,740,176 A	4/1998	Gupta et al.	
5,799,017 A	8/1998	Gupta et al.	
5,818,448 A	10/1998	Katiyar	
5,826,085 A *	10/1998	Bennett et al.	709/227
5,864,542 A	1/1999	Gupta et al.	
5,898,839 A *	4/1999	Berteau	709/202
5,915,090 A *	6/1999	Joseph et al.	709/203
5,920,868 A	7/1999	Fowlow et al.	
5,924,094 A	7/1999	Sutter	
5,958,009 A	9/1999	Friedrich et al.	
5,958,012 A	9/1999	Battat et al.	
5,966,531 A	10/1999	Skeen et al.	
5,993,038 A	11/1999	Sitbon et al.	
6,026,404 A	2/2000	Adunuthula et al.	
6,028,846 A	2/2000	Cain	
6,047,323 A	4/2000	Krause	
6,055,537 A	4/2000	LeTourneau	
6,058,106 A	5/2000	Cudak et al.	
6,067,545 A	5/2000	Wolff	
6,067,577 A	5/2000	Beard	
6,073,163 A *	6/2000	Clark et al.	709/229

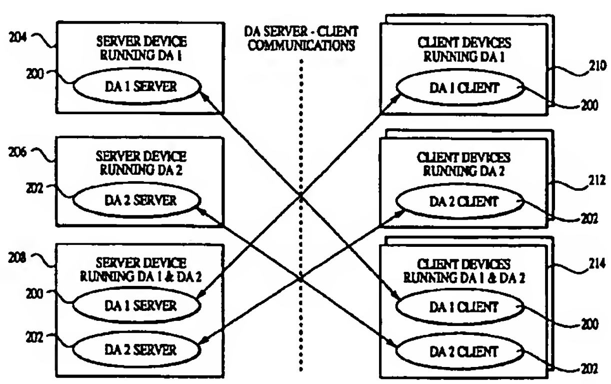
(List continued on next page.)

Primary Examiner—Zarni Maung
Assistant Examiner—Jinsong Hu
(74) *Attorney, Agent, or Firm*—Gazdzinski & Associates

(57) **ABSTRACT**

An improved method and apparatus for balancing distributed applications within a client/server network, such as a cable television network, is disclosed. In one aspect of the invention, a method of balancing the load of distributed application client portions (DACPs) across various server portions (DASPs) and server machines is disclosed. Statistics are maintained by one or more software processes with respect to the available resources of the servers and their loading; new process threads and/or distributed application server portions are allocated across the servers to maintain optimal system performance as client device loading increases or changes. In another aspect of the invention, a novel object-oriented distributed application software architecture employing both vertical and horizontal partitions and "mutable" (i.e., transportable) objects is disclosed. The mutable objects may reside on either the server or client portions of the distributed application while maintaining at least one network partition. A runtime environment adapted for the operation of the foregoing object-oriented distributed application, including an efficient message protocol useful for interprocess communication, is also disclosed. Methods for downloading the DACP from the servers, and scaling the DACP at download based on client device configuration, are further disclosed.

3 Claims, 28 Drawing Sheets



Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L4: Entry 2 of 2 File: USPT Apr 27, 1999

DOCUMENT-IDENTIFIER: US 5898830 A
TITLE: Firewall providing enhanced network security and user transparency

Abstract Text (1):
The present invention, generally speaking, provides a firewall that achieves maximum network security and maximum user convenience. The firewall employs "envoys" that exhibit the security robustness of prior-art proxies and the transparency and ease-of-use of prior-art packet filters, combining the best of both worlds. No traffic can pass through the firewall unless the firewall has established an envoy for that traffic. Both connection-oriented (e.g., TCP) and connectionless (e.g., UDP-based) services may be handled using envoys. Establishment of an envoy may be subjected to a myriad of tests to "qualify" the user, the requested communication, or both. Therefore, a high level of security may be achieved. The usual added burden of prior-art proxy systems is avoided in such a way as to achieve fall transparency-the user can use standard applications and need not even know of the existence of the firewall. To achieve full transparency, the firewall is configured as two or more sets of virtual hosts. The firewall is, therefore, "multi-homed," each home being independently configurable. One set of hosts responds to addresses on a first network interface of the firewall. Another set of hosts responds to addresses on a second network interface of the firewall. In one aspect, programmable transparency is achieved by establishing DNS mappings between remote hosts to be accessed through one of the network interfaces and respective virtual hosts on that interface. In another aspect, automatic transparency may be achieved using code for dynamically mapping remote hosts to virtual hosts in accordance with a technique referred to herein as dynamic DNS, or DDNS.

Detailed Description Text (34):
Referring more particularly to FIG. 4, a load-sharing firewall is realized using a first firewall 407 and a second firewall 408 connected in parallel to a network 420 such as the Internet. Redundancy is provided by conventional DNS procedures. That is, in DNS, redundant name servers are required by the DNS specification. If a query addressed to one of the redundant name servers does not receive a response, the same query may then be addressed to another name server. The same result holds true in FIG. 4. If one of the physical firewall machines 407 or 408 is down, the other machine enables normal operation to continue.

CLAIMS:

9. The method of claim 4, comprising the further steps of, for at least one of the firewalls:

providing multiple physical computers, each configured as a plurality of virtual hosts, a first virtual host on one of said physical machines being identically configured as a second virtual host on another of said physical machines;

wherein said mapping from a name of the second computer to a network address of one of the virtual hosts of the firewall is made dynamically to one of said first virtual host and said second virtual host depending on availability of said one

physical machine and said another physical machine.

Previous Doc Next Doc Go to Doc#